

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1. Sistem**

Menurut Ogata (1996) dalam bukunya yang berjudul “Teknik Kontrol Automatik” sistem itu merupakan kombinasi dari beberapa komponen yang bekerja bersama-sama dan melakukan suatu sasaran tertentu. Sistem tidak dibatasi hanya untuk sistem fisik saja. Konsep sistem dapat digunakan pada gejala yang abstrak dan dinamis.

#### **2.2. Sensor Ultrasonic**

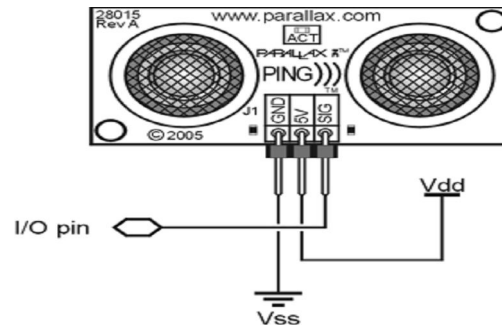
Sensor *ultrasonic* digunakan untuk mengetahui jarak suatu objek. Cara kerja sensor ini dalam mendeteksi objek adalah dengan mengirimkan gelombang *ultrasonic* pendek dan kemudian menunggu pantulan dari gelombang yang dipancarkan tadi kembali ke sensor.(Sumardi, 2013, hal:113)

Didalam kendali mikrokontroler (untuk mengeluarkan pulsa pemicu), sensor mengirimkan gelombang *ultrasonic* pendek dengan frekuensi 40 KHz. Gelombang ini akan melalui udara kira-kira 1130 kaki/ detik, membentur suatu objek dan kemudian kembali ke sensor. Sensor ini menyediakan pulsa keluaran pada mikrokontroler yang akan diteruskan ketika gelombang pantulan terdeteksi oleh sensor.

Sensor ini berfungsi untuk mengubah besaran fisis yang berupa jarak menjadi besaran elektris tegangan. Sensor ini bekerja dengan mengirimkan gelombang *ultrasonic* yang memiliki frekuensi dibawah pendengaran manusia kemudian akan menyediakan pulsa keluaran yang berhubungan langsung dengan waktu yang dibutuhkan bagi pantulan gelombang *ultrasonic* untuk mencapai sensor.

Sensor ini memiliki 3 pin yaitu sebagai berikut :

- a. Pin *Ground*
- b. Pin *Vcc 5 V*
- c. Pin *Input / Output*



**Gambar 2.1** Sensor *Ultrasonic*

### 2.3. Liquid Crystal Display (LCD)

LCD (*Liquid Crystal Display*) adalah salah satu alat yang digunakan sebagai tampilan. LCD menggunakan kristal cair sebagai penampil utama, selain itu LCD juga dapat digunakan untuk menampilkan karakter ataupun simbol. (Sumardi, 2013, hal:36)

LCD memiliki karakteristik sebagai berikut :

- a. Terdapat 16 x 2 karakter huruf yang bisa ditampilkan.
- b. Setiap huruf terdiri dari 5x7 dot matrix + cursor.
- c. Terdapat 192 macam karakter.
- d. Terdapat 80 x 8 bit display RAM (maksimal 80 karakter).
- e. Memiliki kemampuan penulisan dengan 8 bit maupun dengan 4 bit.
- f. Dibangun dengan osilator lokal.
- g. Satu sumber tegangan 5 volt.
- h. Otomatis reset saat tegangan dihidupkan.



**Gambar 2.2** LCD (Sumardi, 2013)

Adapun Fungsi dari masing-masing Pin LCD adalah sebagai berikut :

**Tabel 2.1** Fungsi pin LCD

Pin	Nama	Fungsi
1	VSS	<i>Ground Voltage</i>
2	VCC	+5V
3	VEE	<i>Contrast Voltage</i>
4	RS	<i>Register Select</i> 0 = <i>Instruction Register</i> 1 = <i>Data Register</i>
5	R/W	<i>Read or Write mode</i> 0 = <i>Write mode</i> 1 = <i>Read mode</i>
6	E	<i>Enable</i> 0 = <i>start to lacht data to LCD character</i> 1 = <i>disable</i>
7	DB0	Data bit ke-0 (LSB)
8	DB1	Data bit ke-1
9	DB2	Data bit ke-2
10	DB3	Data bit ke-3
11	DB4	Data bit ke-4
12	DB5	Data bit ke-5
13	DB6	Data bit ke-6
14	DB7	Data bit ke-7
15	BPL	<i>Back Plane Light</i>
16	GND	<i>Ground Voltage</i>

## 2.4. Mikrokontroler

### 2.4.1. Pengertian Mikrokontroler

Mikrokontroler adalah suatu alat elektronika digital yang mempunyai masukan dan keluaran serta kendali program yang bisa ditulis dan di hapus dengan cara khusus. Mikrokontroler adalah mikroprosesor yang dikhususkan untuk instrumentasi dan kendali.(Sumardi, 2013, hal:1)

Mikrokontroler ialah chip yang berisi berbagai unit penting untuk melakukan pemrosesan data (*I/O, timer, memory, Arithmetic Logic Unit (ALU)* dan lainnya) sehingga dapat berlaku sebagai pengendali dan komputer sederhana. (Budiharto, 2011, hal:1)

Mikrokontroler digunakan dalam produk dan alat yang dikendalikan secara otomatis, seperti sistem kontrol mesin, *remote controls*, mesin kantor, peralatan rumah tangga, alat berat, dan mainan. Dengan mengurangi ukuran, biaya, dan konsumsi tenaga dibandingkan dengan mendesain menggunakan mikroprosesor memori, dan alat input output yang terpisah, kehadiran mikrokontroler membuat kontrol elektrik untuk berbagai proses menjadi lebih ekonomis. Dengan penggunaan mikrokontroler ini maka:

1. Sistem elektronik akan menjadi lebih ringkas
2. Rancang bangun sistem elektronik akan lebih cepat karena sebagian besar dari sistem adalah perangkat lunak yang mudah dimodifikasi
3. Pencarian gangguan lebih mudah ditelusuri karena sistemnya yang kompak.

Adapun jenis dari mikrokontroler banyak sekali, di antaranya dari keluarga MCS51 yang diproduksi Atmel, keluarga PIC dari *microchip*.

Intel 8048 adalah mikrokontroler yang pertama dilempar ke pasaran di tahun 1976. Keluarga dari 8048 (MCS-48) adalah 8021, 8022, 8048 dan 8049 yang hingga kini masih digunakan pada alat-alat kedokteran modern dan digunakan pada *keyboard PC* untuk *scanning* tombol.

Generasi kedua mikrokontroler 8 bit adalah keluarga mikrokontroler 8051 (MCS-51). *Chip* ini kemudian dikembangkan menjadi beberapa seri dengan berbagai kemampuan seperti 8031, 80C31, 8051AH, 8751.

Beberapa perusahaan membuat variannya yaitu suatu *chip* yang kompatibel dengan bahasa dan fitur 8051 ditambah dengan kemampuan dan kemudahan khusus. Salah satu perusahaan tersebut adalah ATMEL dengan produknya seperti AT89C51, AT89C2051, AT89S51. Pada *chip-chip* tersebut sudah terdapat *Flash ROM* yang disebut PEROM (*Programable Erasable Read Only Memory*).

Generasi ketiga adalah mikrokontroler 16-bit seri MCS-96 yang dapat melakukan operasi 16-bit dengan kemampuan dan kecepatan proses yang ditingkatkan.

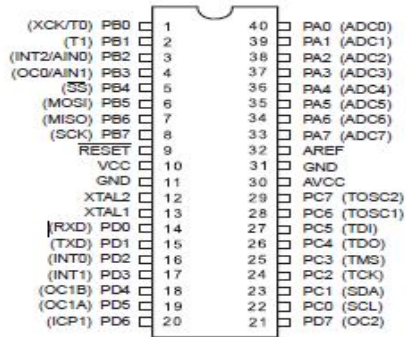
Tetapi meskipun banyak jenisnya, menurut arsitektur prosesornya ada 2, yaitu:

- a. *CISC (Complex Instruction Set Computing)* Prosesor ini mempunyai banyak instruksi tetapi fasilitas internal minim pada seri AT89xx memiliki 255 instruksi).
- b. *RISC (Reduced Instruction Set Computing)* Prosesor ini mempunyai banyak fasilitas internal tetapi jumlah instruksi secukupnya (seri PIC hanya ada sekitar 30-an instruksi). Di atas kertas, dari segi kecepatan RISC memang lebih unggul dibandingkan CISC namun dari segi kinerja sesungguhnya belum tentu. Karena keluarga prosesor RISC hanya menyediakan instruksi untuk fungsifungsi dasar, maka untuk fungsi-fungsi lanjutan yang lebih kompleks, akan diambil alih oleh software, sementara untuk fungsi yang sama, prosesor berbasis CISC dapat memanfaatkan instruksinya sendiri.(Indriyanto dan Cahyono,2010)

#### **2.4.2. Mikrokontroler ATMEGA 32**

Mikrokontroler AVR (*Alf and Vegard's RISC processor*) standar memiliki arsitektur 8 bit, dimana semua instruksi dikemas dalam kode 16 bit dan sebagian besar instruksi dieksekusi dalam satu siklus clock. AVR berteknologi RISC (*Reduced Instruction Set Computing*). AVR dapat dikelompokkan menjadi empat kelas, yaitu keluarga ATtiny, keluarga AT90Sxx, Keluarga ATMEGA, dan keluarga AT86RFxx. (Budiharto, 2011, hal:2)

Mikorkontroler ATMEGA 32 tergolong dalam Mikrokontroler jenis AVR dimana semua instruksi dikemas dalam kode 16 bit dan sebagian besar instruksi dikemas berbeda dengan instruksi MCS51 yang membutuhkan 12 siklus *clock*. ATMEGA 32 memiliki jumlah pin sebanyak 40 buah, dimana 32 pin digunakan untuk keperluan *port I/O*.



**Gambar 2.3** Konfigurasi pin ATMEGA 32

Menurut Budiharto (2011) adapun fungsi-fungsi dari pin mikrokontroler ATMEGA 32 :

- VCC merupakan pin masukan positif catu daya. Setiap perangkat elektronika digital tentunya butuh sumber catu daya yang umumnya sebesar 5 Volt.
- GND sebagai pin *Ground*.
- Port A* (PA0-PA7) merupakan pin I/O dua arah dan dapat diprogram sebagai pin masukan ADC.
- Port B* (PB0-PB7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu *Timer/Counter*, komparator analog, dan SPI.
- Port C* (PC0-PC7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu TWI, Komparator analog dan *Timer* Osilator.
- Port D* (PD0-PD7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu komparator analog, interupsi eksternal, dan komunikasi serial.
- Reset* merupakan pin yang digunakan untuk mereset mikrokontroler.
- XTAL 1 dan XTAL 2 sebagai pin masukan *clock* eksternal. Suatu mikrokontroler membutuhkan sumber detak (*clock*) agar dapat mengeksekusi instruksi yang ada di memori.
- AVCC sebagai pin masukan tegangan untuk ADC.
- AREF sebagai pin masukan tegangan referensi ADC.

Berikut fitur-fitur yang dimiliki oleh mikrokontroler ATMEGA 32 dan ATMEGA 32L:

Kecepatan :

- ATMEGA 32 : 0-16 MHz
- ATMEGA 32L : 0-8 MHz

Catu daya :

- a. ATMEGA 32 : 4.5-5.5 V
- b. ATMEGA 32L : 2.7-5.5 V

Memori :

- a. *Flash Memory* : 32 KB (Daya tahan : 10.000 kali penulisan)
- b. EEPROM : 1024 Byte (Daya tahan : 100.000 kali penulisan)
- c. SRAM Internal : 2 KB

Lama penyimpanan data :

- a. 20 tahun pada suhu 85<sup>0</sup>C
- b. 100 tahun pada suhu 25<sup>0</sup>C

Fitur-fitur peripheral :

- a. 2 buah *timer/counter* 8-bit
- b. Sebuah *timer/counter* 16-bit
- c. 4 buah *channel* PWM
- d. 8 buah *channel* ADC 10-bit
- e. *Interface Two-wire Serial Interface*
- f. *Interface* USART (serial)
- g. *Interface* SPI Master/Slave
- h. *Watchdog timer* yang bisa deprogram
- i. Komparator Analog



**Gambar 2.4** Mikrokontroler ATMEGA 32

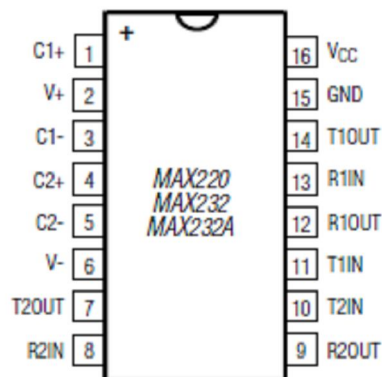
## 2.5. IC MAX232

IC MAX232 dari Maxim Incorporation adalah IC pengubah level RS-232 yang memiliki sebuah *charge pump* yang bisa menghasilkan tegangan +10 V dan -10 V dari tegangan catudaya 5 V. Tegangan-tegangan ini dihasilkan dengan

proses pengisian dan pembuangan 4 buah kapasitor luar yang dihubungkan dengan rangkaian pengganda tegangan internal yang dimiliki oleh IC ini.(Usman, 2008, hal:348)

MAX232 mempunyai dua penerima (RS-232 ke TTL) dan dua pengirim (TTL ke RS-232). IC MAX232 disini berfungsi untuk merubah level tegangan pada COM1 menjadi level tegangan TTL/CMOS.

Berbeda dengan mikrokontroler, dimana level digital untuk *port* serial adalah level TTL (logika 1 dinyatakan sebagai 5 V dan logika 0 sebagai 0 V), *port* serial *PC* menggunakan level RS-232. Dalam RS-232 logika 1 dinyatakan sebagai *Mark* dengan level tegangan antara -3 dan -25 V (negatif), sedangkan logika 0 dinyatakan sebagai *Space* dengan level tegangan antara 3 dan 25 V (positif). Perbedaan level logika ini membuat *port* serial mikrokontroler tidak bisa secara langsung dihubungkan ke *port* serial komputer. Maka dari itu salah satu fungsi IC MAX232 ini adalah sebagai antarmuka antara *port* serial mikrokontroler dan *port* serial komputer.



**Gambar.2.5** IC MAX232

## 2.6. SMS Gateway

### 2.6.1. Pengertian SMS

SMS (*Short Message Service*) merupakan sebuah layanan yang banyak diaplikasikan pada sistem komunikasi tanpa kabel, memungkinkan dilakukannya pengiriman pesan dalam bentuk teks. SMS didukung oleh GSM (*Global System for Mobile Communication*), TDMA (*Time Division Multiple Access*), CDMA (*Code Division Multiple Access*) yang berbasis telepon seluler yang saat ini banyak digunakan. (Wiharto, 2011)



Karakteristik utama SMS adalah SMS merupakan sebuah sistem pengiriman data dalam paket yang bersifat *out-of-band* dengan *bandwidth* kecil. Dengan karakteristik ini, pengiriman suatu *burst* data yang sangat pendek dapat dilakukan dengan efisiensi yang sangat tinggi.

Pada saat mengirim pesan SMS dari *handphone* pesan tersebut tidak langsung dikirimkan ke *handphone* tujuan melainkan dikirimkan terlebih dahulu ke SMS Center baru kemudian diteruskan ke *handphone* tujuan. Dengan adanya SMS Center kita dapat mengetahui status pesan SMS yang dikirim, apabila *handphone* tujuan dalam keadaan aktif dan dapat menerima

Untuk dapat mengirim dan menerima pesan, kita harus melakukan koneksi ke SMS Center. Ada beberapa cara untuk melakukan koneksi ke SMS Center antara lain :

- a. Menggunakan Terminal baik berupa GSM Modem atau Handphone  
Cara ini adalah cara yang paling mudah tetapi memiliki kekurangan antara lain jumlah pesan yang dikirim per menit sangat terbatas.
- b. Koneksi langsung ke SMS Center  
Dengan melakukan koneksi langsung ke SMS Center kita dapat mengirim pesan dengan jumlah yang relatif banyak per menit tergantung dari kapasitas SMS Center itu sendiri.
- c. Menggunakan Software Bantu  
Saat ini banyak vendor telekomunikasi menawarkan *software* bantu untuk melakukan koneksi ke SMS Center, dari yang bersifat *freeware*, *open source* sampai dengan komersial biasanya disesuaikan dengan jumlah pesan SMS yang akan dikirim.

### 2.6.2. Modem Wavecom

Modem Wavecom adalah salah satu jenis *modem* yang dirancang untuk dapat dihubungkan dengan perangkat luar dengan koneksi serial. Modem ini menggunakan format pengiriman data serial biasa sehingga memudahkan saat dihubungkan dengan mikrokontroler. (Melalolin, 2013)

Wavecom adalah pabrikan asal Perancis yaitu Wavecom.SA yang berdiri sejak 1993. Pada awalnya Wavecom merupakan biro konsultan teknologi dan

sistem jaringan nirkabel GSM, dan pada 1996 Wavecom mulai membuat desain daripada modul *wireless* GSM pertamanya dan diresmikan pada tahun 1997, bentuk modul GSM pertama berbasis GSM dengan pengkodean khusus yang disebut *AT-command*. Sulit mencari referensi *module* yang familiar di telinga pengguna Wavecom Indonesia antara lain :

1. Wismo 2C2 atau dikenal juga pembaharunya Wismo Quik Q2303A yang belum mendukung GPRS (masih murni GSM).
2. Wismo Quik Q2403A, mendukung GPRS dan format *AT-command* yang telah berstandar ETSi GSM.
3. Wismo Quik Q2406A
4. Wismo Quik Q2406B (Untuk Eropa tersedia versi uji-coba dengan dukungan *Open AT OS*)
5. Wismo Quik Q24plus, telah mendukung penerapan fungsi *Open AT OS* (kedua di *module* Wavecom yang mendukung *Open AT*).
6. Wismo Quik Q2686/Q2687, pembaharu dari *module* Q24Plus dengan ukuran yang lebih kompak namun sarat fungsi dan integrasi-*Quad Band*.

Selain memproduksi *module* untuk kebutuhan (Machine to Machine), *wireless* telemetri, *wireless telecommunication* dan sebagainya, Wavecom juga memproduksi unit *modem* sendiri yaitu yang saat ini dikenal oleh kita dengan sebutan Wavecom Fastrack. Adapun tipe-tipe dari *modem* Wavecom antara lain adalah :

1. Wavecom Fastrack WMOD (didalam dibenamkan *module* wismo 2C2).
2. Wavecom Fastrack WMOD2 (didalamnya dibenamkan *module* wismo Q2303A).
3. Wavecom Fastrack WMOD3 (didalamnya dibenanmkan *module* wismo Q2303B GPRS).
4. Wavecom Fastrack M1203A (prototipe dari *modem* Fastrack ber*module* wismo Q2403A tanpa fungsi *voice* dan *fax*).
5. Wavecom Fastrack M1203B (jenis massal dari *modem* Fastrack yang dipasarkan di Asia Pasifik dengan dukungan *module* wismo Q2403A dan telah mendukung fasilitas *voice/fax*).

6. Wavecom Fastrack M1206A (jenis prototipe dengan *module* wismo Q2403A dan telah mendukung TCP/IP *stacked* dan syarat telekomunikasi termutakhir kala itu).
7. Wavecom Fastrack M1206B (jenis produksi massal yang paling sukses dengan dilengkapi *module* Q2406A dan pada seri terakhir juga telah di revisi menjadi *module* wismo Q2406B dengan dukungan *voice/fax/mms/tcpip stacked*).
8. Wavecom Fastrack M1306A (jenis pertama dengan revisi ukuran casing dan PCB terbaru yang lebih canggih dari suksesornya M1206B, namun masih bernama *module* Q2406B-non *Open AT*).
9. Wavecom Fastrack M1306B (jenis produksi massal yang paling laris sepanjang masa, dengan dukungan *module* Q2406B (*open AT*) dan Q24Plus classic dengan dukungan penuh terhadap *Open AT* dan TCPIP *Stacked*).
10. Wavecom Fastrack Supreme 10/20 (*module* menggunakan wismo Q2686 dan Q2687 – mendukung penuh fungsi *Open AT* dan *Open IESM port*).
11. Wavecom Fastrack GO (wismo Q2687 dan merupakan *modem* terkecil pertama buatan Wavecom).
12. Wavecom Fastrack XTEND (varian terbaru menggantikan Fastrack Supreme 10 yang telah mendukung EDGE/HSPA).

*Modem* Wavecom Fastrack di Indonesia cukup dikenal dan digunakan untuk mengirim SMS massal hingga sebagai penggerak perangkat elektronik. Beberapa fungsi kegunaan *modem* ini adalah sebagai berikut :

1. *SMS Broadcast Application*
2. *SMS Quiz Application*
3. *SMS Polling*
4. *SMS auto-reply*
5. *M2M integration*

6. Aplikasi *Server* Pulsa
7. Telemetri
8. *Payment Point* Data
9. PPOB



**Gambar 2.6** *Modem Wavecom*

## 2.7. Handphone Pixcom Nano PG 20

Menurut Jamani (2013), Arkanudin (2013), dan Syarmiati (2013) *Handphone* adalah perangkat komunikasi telepon konvensional saluran tetap, namun dapat dibawa kemana-mana. Selain berfungsi untuk melakukan dan menerima panggilan telepon, *handphone* juga berfungsi untuk pengiriman dan penerimaan pesan singkat yang dikenal dengan SMS (*Short Message Service*).

Pixcom salah satu perusahaan *handphone* memiliki suatu produk yaitu Pixcom Nano yang memiliki model yang simpel dan mengedepankan *portability* yang baik untuk memudahkan dalam membawa *handphone* ini kemana saja dan diletakkan dimana saja. Pixcom Nano PG 20 memiliki ukuran yang mungil dan sangat praktis serta tahan lama. Pixcom Nano juga dilengkapi dengan kamera yang digunakan untuk mengambil momen-momen yang anda butuhkan. Selain itu juga Pixcom Nano dilengkapi dengan fitur MP3, Radio, *USB port*, *Micro SD* dan juga dilengkapi dengan baterai *Rechargeable Li-ion polymer battery 3,7 V 600mA*.



**Gambar 2.7** *Handphone Pixcom Nano PG 20*

## **2.8. Pneumatik**

### **2.8.1. Pengertian Pneumatik**

Pneumatik adalah ilmu yang mempelajari gerakan atau perpindahan udara dan gejala atau fenomena udara. Dengan kata lain pneumatik berarti mempelajari tentang gerakan angin (udara) yang dapat dimanfaatkan untuk menghasilkan tenaga dan kecepatan. (Akhmad, 2009)

Menurut Hanwar (2002) dan Feidihal (2002) sistem pneumatik merupakan suatu sistem yang menggunakan tenaga yang disimpan dalam bentuk udara yang dimanfaatkan dan selanjutnya digunakan untuk menghasilkan suatu kerja.

Peralatan sistem pneumatik ini cukup sederhana, dan operatornya memperoleh keamanan dan keselamatan kerja yang lebih terjamin. sistem gerak dalam pneumatik memiliki optimalisasi/efektifitas bila digunakan pada batas waktu tertentu. adapun batas-batas ukuran yang dapat menimbulkan optimalisasi penggunaan pneumatik antara lain : diameter piston antara 6 s/d 320 mm, panjang langkah 1 s/d 2.000 mm, tenaga yang diperlukan 2 s/d 15 bar.

Secara umum komponen-komponen konstruksi pneumatik dapat dikelompokkan menjadi tiga antara lain sebagai berikut :

#### **a. Unit Tenaga (*Power Pack*)**

Unit ini berfungsi untuk membangkitkan tenaga fluida yaitu berupa aliran udara mampat. Unit tenaga ini terdiri atas kompresor yang digerakkan oleh motor listrik atau motor bakar, tangki udara, serta unit pelayanan udara yang terdiri atas filter udara, regulator pengatur tekanan dan lubricator.

b. Unit Pengatur

Unit pengatur merupakan bagian pokok yang menjadikan sistem pneumatik ini termasuk sistem otomasi. Karena dengan unit pengatur ini hasil kerja dari sistem pneumatik dapat diatur secara otomatis baik gerakan, kecepatan, urutan gerak, arah gerakan maupun kekuatannya. Fungsi unit pengatur ini adalah untuk mengatur jalannya fluida hingga menghasilkan usaha yang berupa tenaga mekanik. Unit pengatur ini berupa katup kontrol arah.

c. Unit Penggerak (Aktuator)

Unit ini berfungsi untuk mewujudkan hasil transfer daya dari tenaga fluida, berupa gerakan lurus atau gerakan putar. Penggerak yang menghasilkan gerakan lurus adalah silinder penggerak, sedangkan yang menghasilkan gerakan putar adalah motor pneumatik.

Beberapa keuntungan dalam penggunaan atau penerapan sistem pneumatik, antara lain:

- a. Ketelitian yang tinggi dari peralatan-peralatan pneumatik yang konstruksinya semakin baik memungkinkan suatu pengerjaan yang hampir tidak memerlukan perawatan dalam jangka panjang.
- b. Merupakan media/fluida kerja yang mudah didapat dan mudah diangkut udara dimana saja tersedia dalam jumlah yang tak terhingga.
- c. Udara bertekanan adalah bersih. Kalau ada kebocoran pada saluran pipa, benda-benda kerja maupun bahan-bahan disekelilingnya tidak akan menjadi kotor.
- d. Dapat bertahan lebih baik terhadap keadaan-keadaan kerja tertentu. Udara bersih (tanpa uap air) dapat digunakan sepenuhnya pada suhu-suhu yang tinggi atau pada nilai-nilai yang rendah, jauh di bawah titik beku(masing-masing panas atau dingin).
- e. Aman terhadap kebakaran dan ledakan.
- f. Menguntungkan karena lebih murah dibandingkan dengan komponen-komponen peralatan hidrolik.
- g. Konstruksinya yang kompak dan kokoh.
- h. Memiliki beberapa tekanan kerja sesuai dengan kebutuhan pemakaian.

- i. Dapat dibebani lebih. Pada pembebanan lebih alat-alat udara bertekanan memang akan berhenti tetapi tidak mengalami kerusakan.

### 2.8.2. Prinsip Kerja Pneumatik

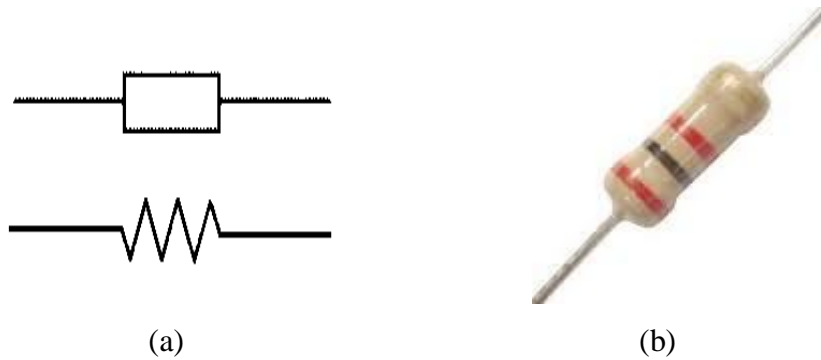
Kompresor diaktifkan dengan cara menghidupkan penggerak awal umunya motor listrik. Udara akan disedot oleh kompresor kemudian ditekan ke dalam tangki udara hingga mencapai tekanan beberapa bar. Untuk menyalurkan udara bertekanan ke seluruh sistem (sirkuit pneumatik) diperlukan unit pelayanan atau *service unit* yang terdiri dari penyaring (*filter*), katup kran (*shut off valve*) dan pengatur tekanan (*regulator*). Cara ini diperlukan karena udara bertekanan yang diperlukan di dalam sirkuit pneumatik harus benar-benar bersih, tekanan operasional pada umumnya hanyalah sekitar 6 bar. Selanjutnya udara bertekanan disalurkan dengan bekerjanya *solenoid valve pneumatic* ketika mendapat tegangan input pada kumparan dan menarik *plunger* sehingga udara bertekanan keluar dari *outlet port* melalui selang elastis menuju katup pneumatik (katup pengarah/ *inlet port pneumatic*). Udara bertekanan yang masuk akan mengisi tabung pneumatik dan membuat piston bergerak maju dan udara bertekanan tersebut harus mendorong piston dan akan berhenti di lubang *outlet port* pneumatik atau batas dorong piston.



**Gambar 2.8** Silinder Pneumatik

## 2.9. Resistor

Resistor merupakan suatu benda yang dibuat sebagai penghambat atau penahanan arus listrik yang mengalir pada suatu rangkaian dengan tujuan untuk mengatur arus yang mengalir yang dinyatakan dengan satuan *ohm*. (Budiman, 1992, hal:207)



**Gambar 2.9** (a) Simbol Resistor dan (b) Resistor

Gelang yang terdapat dalam resistor sebenarnya merupakan kode dari nilai resistansi yang terkandung di dalamnya, untuk dapat membacanya kita harus mengetahui kode tersebut. Berikut adalah kode warna resistor:

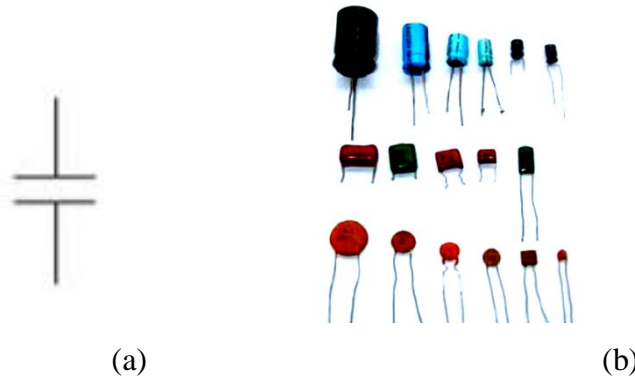
**Tabel 2.2** Kode warna resistor

Warna	Gelang 1	Gelang 2	Gelang 3	Multiplier	Toleransi
				Gelang 4	Gelang 5
Hitam	0	0	0	1 $\Omega$	
Coklat	1	1	1	10 $\Omega$	$\pm 1 \%$
Merah	2	2	2	100 $\Omega$	$\pm 2 \%$
Orange	3	3	3	1 K $\Omega$	
Kuning	4	4	4	10 K $\Omega$	
Hijau	5	5	5	100 K $\Omega$	$\pm 0,5 \%$
Biru	6	6	6	1 M $\Omega$	$\pm 0,25 \%$
Ungu	7	7	7	10 M $\Omega$	$\pm 0,10 \%$
Abu-abu	8	8	8		$\pm 0,05 \%$
Putih	9	9	9		
Emas				0,1 $\Omega$	$\pm 5 \%$
Perak				0,01 $\Omega$	$\pm 10 \%$



### 2.10. Kapasitor

Kapasitor merupakan suatu alat elektronis yang terdiri dari konduktor dan insulator yang mempunyai sifat sebagai penyimpan muatan listrik.(Budiman, 1992, hal:45)



**Gambar 2.10** (a) Simbol Kapasitor dan (b) Kapasitor

### 2.11. Dioda

Dioda merupakan suatu komponen elektronik yang terdiri dari dua buah elektroda (yaitu *anoda* dan *katoda*) yang digunakan untuk meratakan atau mengarahkan aliran ke satu jurusan, yaitu dari anoda menuju katoda. Bahan untuk dioda yang umum digunakan yaitu *silikon* (Si) dan *Germanium*(Ge) dimana kedua bahan tersebut merupakan bahan semi konduktor.(Budiman, 1992, hal:72)



**Gambar 2.11** (a) Simbol Dioda dan (b) Dioda

### 2.12. Transistor

Menurut Budiman (1992) Transistor berasal dari dua buah perkataan yaitu dari kata *transfer* dan resistor. *Transfer* berarti pemindahan dan resistor berarti penahanan. Jadi transistor adalah pemindahan penahanan. Elektroda-elektroda pada transistor terdiri dari basis, kolektor dan emitor. Elektroda-elektroda ini cukup





**Gambar 2.14** *Relay*

## **2.15. Bahasa C**

Pencipta bahasa C adalah Brian W. Kernighan dan Dennis M. Ritchie pada sekitar tahun 1972. C adalah bahasa pemrograman terstruktur, yang membagi program dalam bentuk sejumlah blok. Tujuannya adalah untuk memudahkan dalam pembuatan dan pengembangan program. Program yang ditulis dengan menggunakan C mudah sekali untuk dipindahkan dari satu jenis mesin ke jenis mesin lainnya. Hal ini berkat adanya standarisasi bahasa C yaitu berupa standar ANSI (*American National Standards Institute*) yang dijadikan acuan oleh para pembuat kompilasi C. (Bejo, 2008)

### **2.15.1. Bentuk Dasar Program C**

Sebuah program dalam bahasa C setidaknya memiliki sebuah fungsi. Fungsi ini disebut dengan fungsi utama (fungsi *main*) dan memiliki kerangka program sebagai berikut :

```
Void main (void)
{
    //pernyataan
}
```

Jika kita mempunyai beberapa fungsi yang lain maka fungsi utama inilah yang memiliki kedudukan tertinggi dibandingkan fungsi-fungsi yang lain sehingga setiap kali program dijalankan akan selalu dimulai dari pemanggilan fungsi utama. Fungsi-fungsi yang lain akan dipanggil setelah fungsi utama dan dijalankan melalui pernyataan-pernyataan yang berada dalam fungsi utama.

### 2.15.2. Pengenal

Pengenal (*identifier*) merupakan sebuah nama yang didefinisikan oleh pemrogram untuk menunjukkan identitas dari sebuah konstanta, variabel, fungsi, label, atau tipe data khusus. Pemberian nama sebuah pengenal ditentukan bebas sesuai keinginan pemrogram tetapi harus memenuhi aturan berikut:

1. Karakter pertama tidak boleh menggunakan angka.
2. Karakter kedua dapat berupa huruf, angka, atau garis bawah.
3. Bersifat *Case Sensitive*, yaitu huruf kapital dan huruf kecil dianggap berbeda.
4. Tidak boleh menggunakan kata-kata yang merupakan sintaks maupun operator dalam pemrograman C, misalnya : *void, short, const, if, bit, long, case, do, switch, char, float, for, else, break, int, double, include, while*.

### 2.15.3. Tipe Data

Tipe data merupakan bagian program yang paling penting karena tipe data mempengaruhi setiap instruksi yang akan dilaksanakan oleh komputer. Pemilihan tipe data yang tepat akan membuat proses operasi data menjadi lebih efisien dan efektif. Adapun tipe-tipe data yang ada dalam bahasa C adalah sebagai berikut:

**Tabel 2.3** Tipe data

Tipe Data	Ukuran	Jangkauan nilai
<i>Bit</i>	1 bit	0 atau 1
<i>Char</i>	1 byte	-128 s/d 127
<i>Unsigned Char</i>	1 byte	0 s/d 255
<i>Signed Char</i>	1 byte	-128 s/d 127
<i>Int</i>	2 byte	-32.768 s/d 32.767
<i>Short Int</i>	2 byte	-32.768 s/d 32.767
<i>Unsigned Int</i>	2 byte	0 s/d 65.535
<i>Signed Int</i>	2 byte	-32.768 s/d 32.767
<i>Long Int</i>	4 byte	-2.147.483.648 s/d 2.147.483.647
<i>Unsigned Long Int</i>	4 byte	0 s/d 4.294.967.295

<i>Signed Int</i>	4 byte	-2.147.483.648 s/d 2.147.483.647
<i>Float</i>	4 byte	$1.2 \times 10^{-38}$ s/d $3.4 \times 10^{+38}$
<i>Double</i>	4 byte	$1.2 \times 10^{-38}$ s/d $3.4 \times 10^{+38}$

#### 2.15.4. Konstanta dan Variabel

Konstanta dan variabel merupakan sebuah tempat untuk menyimpan data yang berada di dalam memori. Konstanta berisi data yang nilainya tetap dan tidak dapat diubah selama program dijalankan, sedangkan variabel berisi data yang bisa berubah nilainya pada saat program dijalankan. Untuk membuat sebuah konstanta atau variabel maka kita harus mendeklarasikannya terlebih dahulu yaitu dengan sintaks sebagai berikut.

##### 1. Deklarasi Konstanta :

```
Const [tipe_data] [nama_konstanta]=[nilai]
```

Contoh :

```
Const char konstan=0x10;
```

##### 2. Deklarasi Variabel :

```
[tipe_data] [nama_variabel]=[nilai_awal];
```

Contoh :

```
Char variabelku;  
Char variabelkku=0x20;  
Bit variabel_bit;  
Bit variabel_bit=1;
```

#### 2.15.5. Pernyataan IF

Pernyataan *if* digunakan untuk melakukan pengambilan keputusan terhadap dua buah kemungkinan yaitu mengerjakan suatu blok pernyataan atau tidak. Bentuk pernyataan *if* adalah sebagai berikut :

```
If (kondisi)  
{  
    // blok pernyataan yang akan dikerjakan  
    // jika kondisi if terpenuhi  
}
```

Contoh :

```
If (PINA>0x80)
{
Dataku = PINA; PORTC=0xFF;
}
```

Pernyataan *if* diatas akan mengecek apakah data yang terbaca pada *port A* (PINA) nilainya lebih dari 0x80 atau tidak, jika ya maka variabel dataku diisi dengan nilai PINA dan data 0xFF dikeluarkan ke *port C*.

Apabila dalam blok pernyataan hanya terdapat satu pernyataan saja maka tanda { dan } dapat dihilangkan seperti pada contoh berikut :

```
If (PINA>0x80)
PORTC=0xFF
```

### 2.15.6. Pernyataan Switch

Pernyataan *Switch* digunakan untuk melakukan pengambilan keputusan terhadap banyak kemungkinan. Bentuk pernyataan *Switch* adalah sebagai berikut :

```
switch (ekspresi) {
    Case nilai_1 : pernyataan_1 ; break;
    Case nilai_2 : pernyataan_1 ; break;
    Case nilai_3 : pernyataan_1 ; break;
    Default : pernyataan_default; break;
}
```

Pada pernyataan *switch*, masing-masing pernyataan (pernyataan\_1 sampai dengan pernyataan\_default) dapat berupa satu atau beberapa perintah dan tidak perlu berupa blok pernyataan. Pernyataan\_1 akan dikerjakan jika ekspresi bernilai sama dengan nilai\_1, pernyataan\_2 akan dikerjakan jika ekspresi bernilai sama dengan nilai\_2, pernyataan\_3 akan dikerjakan jika ekspresi bernilai sama dengan nilai\_3, dan seterusnya. Pernyataan\_default bersifat opsional, artinya boleh ada boleh tidak. Jika ada maka pernyataan\_default akan dikerjakan jika nilai ekspresi tidak ada yang sama satupun dengan salah satu nilai\_1, nilai\_2, nilai\_3 dan seterusnya.

Setiap akhir pernyataan harus diakhiri dengan *break*, karena ini digunakan untuk keluar dari pernyataan *switch*. Contoh :

```
switch (PINA)
{
    Case 0xFE : PORTC = 0x00; break;
```

```

        Case 0xFD : PORTC = 0xFF; break;
    }

```

Pernyataan diatas berarti membaca *port A*, kemudian datanya (PIN<sub>A</sub>) akan dicocokkan dengan nilai case. Jika PIN<sub>A</sub> bernilai 0xFE maka data 0x00 akan dikeluarkan ke *port C* kemudian program keluar dari pernyataan *switch* tetapi jika PIN<sub>A</sub> bernilai 0xFD maka data 0xFF akan dikeluarkan ke *port C* kemudian program keluar dari pernyataan *switch*.

### 2.15.7. Pernyataan While

Pernyataan *while* digunakan untuk perulangan sebuah pernyataan atau blok pernyataan yang secara terus menerus selama kondisi tertentu masih terpenuhi. Bentuk pernyataan *while* adalah sebagai berikut :

```

While (kondisi) {
    // sebuah pernyataan atau blok pernyataan
}

```

Jika pernyataan yang akan diulang hanya berupa sebuah pernyataan saja maka tanda { dan } bisa dihilangkan. Contoh :

```

Unsigned char a=0;
...
While (a<10) {
    PORTC=a;
    a++;
}

```

Pernyataan diatas akan mengeluarkan data *a* ke *port C* secara berulang-ulang. Setiap kali pengulangan nilai *a* akan bertambah 1 dan setelah nilai *a* mencapai 10 maka pengulangan selesai.

### 2.15.8. Pernyataan For

Pernyataan *for* digunakan untuk melakukan pengulangan sebuah pernyataan atau blok pernyataan, tetapi berapa kali jumlah pengulangannya dapat ditentukan secara lebih spesifik. Bentuk pernyataannya adalah sebagai berikut :

```

for (nilai_awal; kondisi; perubahan) {
    // sebuah pernyataan atau blok pernyataan
}

```

Nilai\_awal merupakan nilai inisial awal sebuah variabel yang didefinisikan terlebih dahulu untuk menentukan nilai variabel pertama kali sebelum pengulangan.

Kondisi merupakan pernyataan pengetesan untuk mengontrol pengulangan, jika pernyataan kondisi terpenuhi maka blok pernyataan akan diulang terus sampai pernyataan kondisi tidak terpenuhi.

Perubahan adalah pernyataan yang digunakan untuk melakukan perubahan nilai variabel baik naik maupun turun setiap kali pengulangan dilakukan.

Contoh ;

```
Unsigned int a;
For (a =1, a<10, a++) {
PORTC=a;
}
```

Pertama kali nilai a adalah 1, kemudian data a dikeluarkan ke *port C*. selanjutnya data a dinaikkan (a++) jika kondisi a <10 masih terpenuhi maka data a akan terus dikeluarkan ke *port C*.

## 2.16. Code Vision AVR

Code Vision AVR *compiler* merupakan *compiler* bahasa C untuk AVR. Kompiler ini cukup memadai untuk belajar AVR, karena mudah penggunaannya juga didukung berbagai fitur yang sangat membantu dalam pembuatan *software* untuk keperluan pemrograman AVR. (Sumardi, 2013, hal:12)

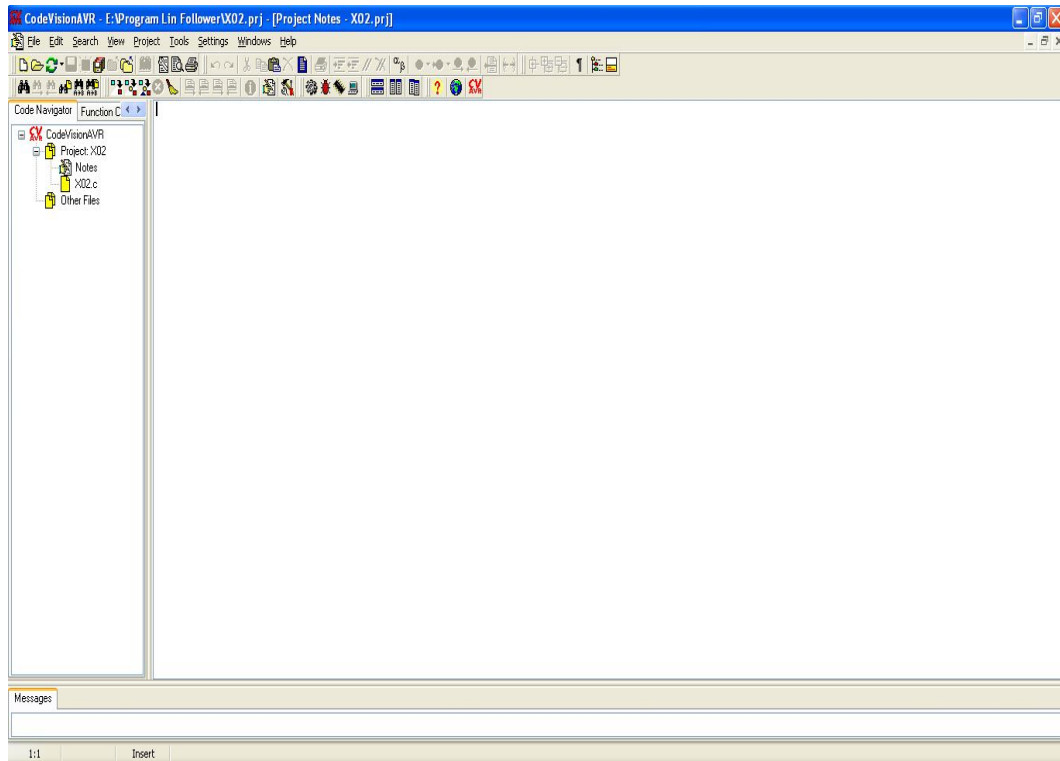
CVAVR dapat berjalan dibawah sistem operasi Windows 9x, Me, NT, 2000 dan XP. CVAVR ini dapat mengimplementasikan hampir semua instruksi bahasa C yang sesuai dengan arsitektur AVR, bahkan terdapat beberapa keunggulan tambahan untuk memenuhi keunggulan spesifik dari AVR. Hasil kompilasi objek CVAVR bisa digunakan sebagai *source debug* dengan AVR Studio debugger dari ATMEL.

Selain pustaka standar bahasa C, CVAVR juga menyediakan pustaka tambahan yang sangat membantu pemrograman AVR, yaitu :

- a. Alphanumeric LCD *modules*
- b. Philips 12C *bus*
- c. National *semiconductor* LM75 *temperature sensor*



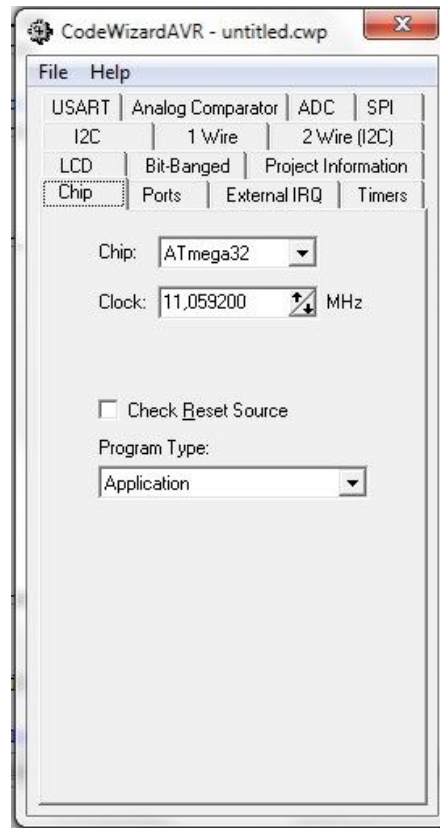
- d. Philips PCF8563, PCF8583, Maxim/Dallas *semiconductor* DS1302 and DS1307 *real time clocks*
- e. Maxim/Dallas *semiconductor* 1 *wire protocol*
- f. Maxim/Dallas *semiconductor* DS 1820, DS28S20, DS18B20 *Temperature sensor*
- g. Maxim/Dallas *semiconductor* DS 1621 *Thermometer/Thermostat*
- h. Maxim/Dallas *semiconductor* DS2430 and DS2433 *EEPROMs*
- i. *SPI*
- j. *Power Management*
- k. *Delays*
- l. *Gray Code Conversion*



**Gambar 2.15** Tampilan *Interface* CodeVision AVR (Sumardi, 2013)

Gambar di atas menunjukkan jendela *interface* dari program CodeVisionAVR, dimana pada *interface* tersebut terbagi dalam beberapa bagian yaitu *toolbar*, *messages*, *navigator* dan jendela tempat penulisan program.

CodeVisionAVR menyediakan suatu fasilitas yang bernama Code Wizard AVR, dimana fasilitas ini mempermudah dalam pemilihan jenis mikrokontroler, serta pengaktifan fasilitas-fasilitas dari mikrokontroler seperti *timer*, *LCD*, *input/output*, *external IRQ*, dan lain-lainnya. Gambar di bawah ini menunjukkan fasilitas dari Code Wizard AVR.






**Gambar 2.16** Code Wizard AVR

Keterangan *tool* pada *toolbar* CodeVisionAVR dapat dilihat pada tabel berikut :

**Tabel 2.4** Keterangan *tool* pada *toolbar* CodeVisionAVR

Icon	Nama	Fungsi
	<i>Create New File</i>	Membuat <i>file</i> baru
	<i>Open File</i>	Menbuka <i>file</i>
	<i>Print File</i>	Mencetak <i>file</i>

	<i>Run the CodeWizardAVR</i>	Menjalankan fasilitas CodeWizardAVR
	<i>Check Syntax</i>	Memeriksa kesalahan penulisan bahasa pemrograman
	<i>Run the Chip Programmer</i>	Menjalankan fasilitas Chip Programmer
	<i>Build the Project</i>	Men-compile program

### 2.17. USBasp Programmer

USBasp *Programmer* merupakan USB yang termasuk dalam sirkuit *programmer* atau yang lebih dikenal sebagai *downloader* untuk *chip* mikrokontroler tipe AVR Atmel.

USBasp *Programmer* ini merupakan *open source hardware* yang berarti *design* skematik dan *layoutnya* yang bisa kita lihat dan buat *prototypenya* sendiri, begitu juga dengan program *bootloader* yang tertanam dalam chip. *Programer* ini hanya menggunakan *firmware* USB *driver* dan tidak diperlukan kontroler khusus. USBasp umumnya dipergunakan untuk melakukan pemrograman mikrokontroler Atmel AVR. Termasuk yang sudah umum dipergunakan di Indonesia seperti ATTiny 2313, ATMEGA 8, ATMEGA 8535, ATMEGA 16 dan ATMEGA 32.

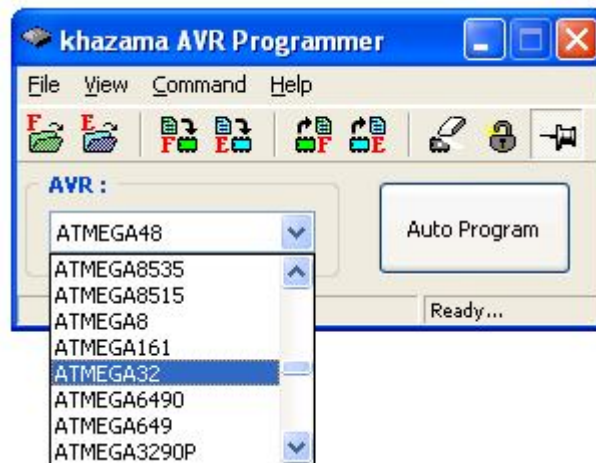


**Gambar 2.17** USBasp *Programer*

### 2.18. Khazama AVR

Menurut Sanjaya (2007) Khazama AVR merupakan suatu aplikasi yang digunakan sebagai proses *download* program ke dalam mikrokontroler. Untuk menggunakan Khazama AVR *Programmer* kita terlebih dahulu memilih

mikrokontroler yang akan digunakan. Cara pemilihannya dapat dilihat seperti pada gambar 2.18.



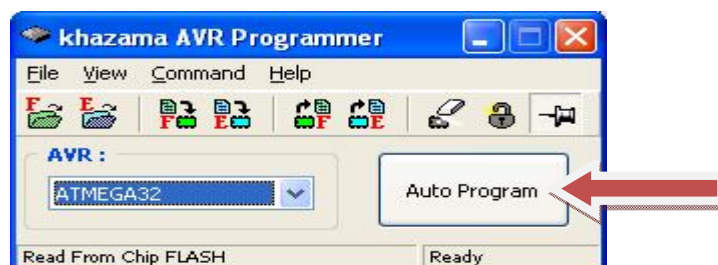
**Gambar 2.18** Pemilihan mikrokontroler pada Khazama AVR

Setelah kita memilih mikrokontroler yang kita gunakan kemudian kita memilih *file* yang berekstensi “.Hex” yang akan kita *download* ke dalam mikrokontroler. Untuk memilih *file* maka klik pada bagian gambar dibawah ini kemudian carilah *file* tersebut di *folder* yang anda simpan.



**Gambar 2.19** Cara mencari *file* .Hex pada Khazama AVR

Apabila *file* yang berekstensi “.Hex “ tadi telah ditemukan maka tekan “*Open*” lalu kemudian kita klik “*Auto Program*” maka proses *download* ke dalam mikrokontroler akan dilaksanakan.

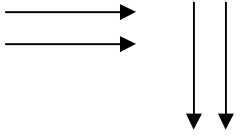
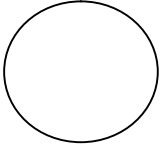
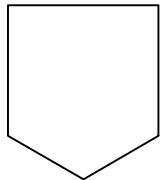



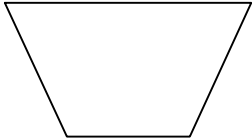
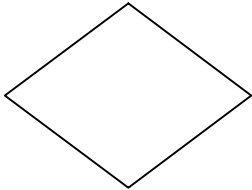
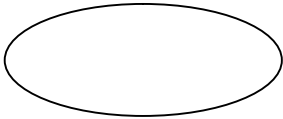


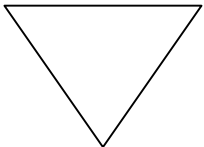

**Gambar 2.20** Pemilihan *Auto Program* pada Khazama AVR

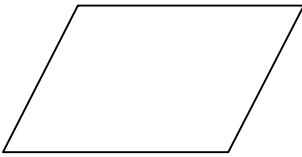
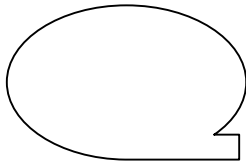
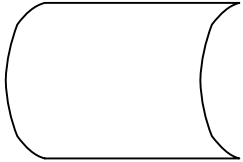

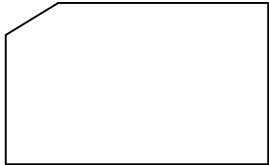
### 2.19. Flowchart

Menurut Kristanti (2012) dan Redita (2012) *Flowchart* adalah cara penyajian *visual* aliran data melalui sistem informasi, *Flowchart* dapat membantu menjelaskan pekerjaan yang saat ini dilakukan dan bagaimana cara meningkatkan atau mengembangkan pekerjaan tersebut. Dengan menggunakan *flowchart* dapat juga membantu untuk menemukan elemen inti dari sebuah proses, selama garis digambarkan secara jelas antara di mana suatu proses berakhir dan proses selanjutnya dimulai. Adapun simbol-simbol dari *flowchart* adalah sebagai berikut:

**Tabel 2.5** Tabel simbol diagram flowchart.

No	Simbol	Keterangan
1		Simbol arus / <i>flow</i> , yaitu menyatakan jalannya arus suatu proses
2		Simbol <i>connector</i> , berfungsi menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama
3		Simbol <i>offline connector</i> , menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda
4		Simbol <i>process</i> , yaitu menyatakan suatu tindakan (proses) yang dilakukan oleh komputer

5		Simbol <i>manual</i> , yaitu menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer
6		Simbol <i>decision</i> , yaitu menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban : ya / tidak
7		Simbol <i>terminal</i> , yaitu menyatakan permulaan atau akhir suatu program
8		Simbol <i>predefined process</i> , yaitu menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal
9		Simbol <i>keying operation</i> , menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai keyboard
10		Simbol <i>offline-storage</i> , menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu
11		Simbol <i>manual input</i> , memasukkan data secara manual dengan menggunakan online keyboard

12		Simbol <i>input/output</i> , menyatakan proses input atau output tanpa tergantung jenis peralatannya
13		Simbol <i>magnetic tape</i> , menyatakan input berasal dari pita magnetis atau output disimpan ke pita magnetis
14		Simbol <i>disk storage</i> , menyatakan input berasal dari disk atau output disimpan ke disk
15		Simbol <i>document</i> , mencetak keluaran dalam bentuk dokumen (melalui printer)
16		Simbol <i>punched card</i> , menyatakan input berasal dari kartu atau output m ditulis ke kartu